



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification<sup>6</sup> :  
H04L 12/56, H04Q 11/04

A1

(11) International Publication Number: WO 98/06203

(43) International Publication Date: 12 February 1998 (12.02.98)

(21) International Application Number: PCT/GB97/02118

(22) International Filing Date: 5 August 1997 (05.08.97)

(30) Priority Data:  
08/692,252 5 August 1996 (05.08.96) US

(71) Applicant: ADVANCED TELECOMMUNICATIONS MODULES LTD. [GB/GB]; Mount Pleasant House, 2 Mount Pleasant, Huntingdon Road, Cambridge CB3 0BL (GB).

(72) Inventors: MOORE, Mark, Justin; Bentinck Cottage, 3 Bentinck Street, Cambridge CB2 1HG (GB). STARK, Gavin, J.; 86 Arbury Road, Cambridge CB4 2GF (GB).

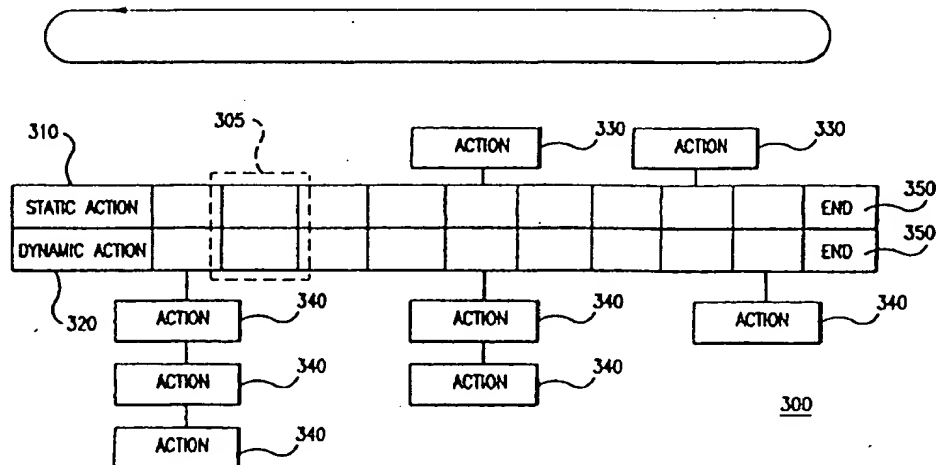
(74) Agents: McCALLUM, William, Potter et al.; Cruikshank & Fairweather, 19 Royal Exchange Square, Glasgow G1 3AE (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.

(54) Title: METHOD AND APPARATUS FOR SOURCE RATE PACING IN AN ATM NETWORK



(57) Abstract

A method and apparatus for scheduling the transmission of cells onto an ATM network, or other packet switching network, is disclosed. The central feature of the scheduling mechanism is a multi-functional timing ring which accommodates both preallocated static scheduling for use with CBR and real-time VBR virtual circuits, and dynamic scheduling for use with ABR, UBR, and non-real time VBR virtual circuits. The timing ring, in which cell transmissions are defined as actions, is processed sequentially in a burst fashion. Static actions are always performed at their allocated time intervals, so fixed transmission intervals can be guaranteed for CBR and real-time VBR traffic. Dynamic actions are moved from the timing ring to a latent queue, which permits dynamic actions to be performed during their scheduled time slot or during the first available time slot thereafter. This mechanism permits ABR and non-real-time VBR traffic contracts to be maintained.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## METHOD AND APPARATUS FOR SOURCE RATE PACING IN AN ATM NETWORK

### 5 Technical Field

The present invention relates generally to data communication networks and, more particularly, to transmission rate control mechanisms for use in asynchronous transfer mode networks.

### 10 Background of the Invention

Asynchronous Transfer Mode ("ATM") is an emerging packet switching network technology designed to provide service for a wide variety of applications such as voice, video and data. Originally proposed for use in the Broadband Integrated Services Digital Network ("B-ISDN") by the International Telegraph and Telephone Consultative Committee ("CCITT"), now reorganized as the Telecommunications Standardization Sector of the International Telecommunication Union ("ITU-T"), ATM is presently moving beyond the wide area network setting into the private network arena as a platform for local area networks ("LANs") with multimedia capabilities. ATM is now well known in the art and is described in various references. *E.g.*, Martin de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN* (2nd Ed., Ellis Horwood Ltd., West Sussex, England, 1993).

In an ATM network, as defined by the CCITT standards, information is carried in packets of fixed size, specified for B-ISDN as 53 bytes or octets, called cells. These cells are individually labelled by addressing information contained in the first 5 bytes (octets) of each cell. Although ATM evolved from Time Division Multiplexing concepts, cells from multiple sources are statistically multiplexed into a single transmission facility. Cells are identified by the contents of their headers rather than by their time position in the multiplexed stream. A single ATM transmission facility may carry hundreds of thousands of ATM cells per second originating from a multiplicity of sources and travelling to a multiplicity of destinations.

ATM is a connection-oriented technology. Rather than broadcasting cells onto a shared wire or fiber for all network members to receive, a specific routing path through the

network, called a virtual circuit, is set up between two end nodes before any data is transmitted. Cells identified with a particular virtual circuit are delivered to only those nodes on that virtual circuit.

5 The backbone of an ATM network consists of switching devices capable of handling the high-speed ATM cell streams. The switching components of these devices, commonly referred to as the switch fabric, perform the switching function required to implement a virtual circuit by receiving ATM cells from an input port, analyzing the information in the header of the incoming cells in real-time, and routing them to the appropriate destination port. Millions of cells per second need to be switched by a single device.

10 Importantly, this connection-oriented scheme permits an ATM network to guarantee the minimum amount of bandwidth required by each connection. Such guarantees are made when the connection is set-up. When a connection is requested, an analysis of existing connections is performed to determine if enough total bandwidth remains within the network to service the new connection at its requested capacity. If the necessary bandwidth  
15 is not available, the connection is refused.

In order to achieve efficient use of network resources, bandwidth is allocated to established connections under a statistical multiplexing scheme. Therefore, congestion conditions may occasionally occur within the ATM network resulting in cell transmission delay or even cell loss. To ensure that the burden of network congestion is placed upon  
20 those connections most able to handle it, ATM offers multiple grades of service. These grades of service support various forms of traffic requiring different levels of cell loss probability, transmission delay, and transmission delay variance, commonly known as delay jitter. It is known, for instance, that many multimedia connections, *e.g.*, video streams, can tolerate relatively large cell losses, but are very sensitive to delay variations from one  
25 cell to the next. In contrast, traditional forms of data traffic are more tolerant of large transmission delays and delay variance, but require very low cell losses. This variation in requirements can be exploited to increase network performance.

The ATM Forum, an organization of networking vendors, has standardized four of these grades of service, or classes of virtual circuits: constant bit rate ("CBR") circuits,  
30 variable bit rate ("VBR") circuits, available bit rate ("ABR") circuits and unspecified bit

rate ("UBR") circuits. These four categories define the qualities of service available to a particular connection, and are selected when a connection is established.

A CBR virtual circuit is granted a permanent allocation of bandwidth along its entire path. The sender is guaranteed a precise time interval, or fixed rate, to send data,  
5 corresponding to the needed bandwidth, and the network guarantees to transmit this data with minimal delay and delay jitter. A CBR circuit is most appropriate for real-time video and audio multimedia streams which require network service equivalent to that provided by a synchronous transmission network. From the perspective of the source and destination, it must appear as if a virtual piece of wire exists between the two points. This requires that  
10 the transmission of each cell belonging to this data stream occur at precise intervals.

A VBR virtual circuit is initially specified with an average bandwidth and a peak cell rate. This type of circuit is appropriate for high priority continuous traffic which contains some burstiness, such as compressed video streams. The network may "overbook" these connections on the assumption that not all VBR circuits will be handling  
15 traffic at a peak cell rate simultaneously. However, although the transmission rate may vary, applications employing VBR service often require low delay and delay jitter.

An ABR virtual circuit, sometimes referred to as connection-oriented data traffic, is appropriate for network connections with uncertain requirements for throughput and delay, such as general data traffic. Currently, ABR circuits are specified with a bandwidth  
20 range, defined by a minimum bandwidth and a maximum bandwidth. It is contemplated that the actual transmission bandwidth within this range will be dynamically negotiated with the network using some form of flow control. See Flavio Bonomi and Kerry W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Network*, March/April 1995, pp. 25-39.

25 A UBR virtual circuit, sometimes referred to as connectionless data traffic, is employed for the lowest priority data transmission; it has no specified associated bandwidth. The sender may send its data as it wishes, but the network makes no guarantee that the data will arrive at its destination within any particular time frame. This service is intended for applications with minimal service requirements, e.g., file transfers submitted in the  
30 background of a workstation.

A particular end-node on the network may have many virtual circuits of these varying classes open at any one time. The network interface at the end-node is charged with the task of scheduling the transmission of cells from each of these virtual circuits in some ordered fashion. At a minimum, this will entail pacing of cells from CBR circuits  
5 at a fixed rate to achieve virtual synchronous transmission. Additionally, some form of scheduling may be implemented within some or all of the switches which form the ATM network. Connections which have deviated from their ideal transmission profile as a result of anomalies in the network can be returned to an acceptable service grade.

At least two concerns are implicated in this scheduling. First, CBR traffic must be  
10 transferred at the necessary fixed rate with little or no delay jitter. Second, ideally VBR and ABR traffic should be transmitted in a manner as near as possible to their defined transfer patterns and no connection should be permitted to exceed its allocated bandwidth or peak cell rate. Although ATM networks possess other mechanisms for dealing with congestion control, adherence to the traffic contract by a source node is an important factor  
15 in overall network efficiency. See Dimitri Bertsekas & Robert Gallager, *Data Networks*, pp. 138-39 (2nd Ed., Prentice Hall, Englewood Cliffs, New Jersey, 1992).

Prior art systems exist which handle the relatively simple task of scheduling CBR traffic for transmission onto an ATM network, but these systems typically perform no scheduling of VBR and ABR traffic. For example, United States Patent No. 5,390,184 to  
20 Morris discloses a scheduling mechanism for use within an ATM switch. CBR traffic is scheduled using a circular buffer containing slots corresponding to the time to transmit a single cell. Unlike VBR or ABR circuits, the bandwidth, and thus the proper cell pacing, of CBR circuits is predefined. Therefore, slots in the circular buffer are statically reserved when a CBR connection is established. The scheduler advances through the circular buffer  
25 transmitting CBR cells when a reserved slot is detected. No scheduling of other forms of traffic is performed, rather waiting VBR/ABR/UBR cells are simply transmitting during time slots not allocated for CBR traffic. Systems such as these achieve proper pacing of synchronous data streams, but do not serve to maintain VBR/ABR data streams in compliance with their negotiated traffic contracts.

30 The lack of a scheduling mechanism for VBR and ABR circuits within an end-node network interface, places a burden upon applications themselves to generate data at

appropriate rates for the network implementation. Requiring applications to perform pacing of VBR/ABR streams may be permissible when such applications have been designed with an ATM network connection in mind. However, such solutions are increasingly unacceptable as ATM moves into the LAN setting. Applications designed for conventional LAN technologies cannot be conveniently adapted to produce data under defined pacing conditions; rather, these applications are likely to transmit data without concern for network congestion conditions. If no pacing of these streams is performed, congestion caused by bursty VBR/ABR streams may impair network performance unless the network itself takes other measures to police transmissions. Of course, a simple method to prevent such network congestion is to request an amount of bandwidth much greater as that which can be reasonably expected to be required at any time by a particular application. This solution, however, leads to an underutilization of the network and an inefficient allocation of resources.

Therefore, there persists a need for a cell pacing mechanism capable of producing uniform transmission of ABR, VBR and UBR data streams, as well as CBR streams. Such a system would provide fixed rate pacing of CBR streams while simultaneously policing and shaping VBR, ABR and UBR data traffic.

#### Summary of the Invention

The present invention relates to transmission apparatus for scheduling the transmission of packets onto an ATM, or other packet switching, network. The apparatus comprises a timing ring containing entries dynamically allocated for the transmission of packets of virtual circuits, typically ABR, UBR and non-real time VBR circuits. An entry on the timing ring represents an available time slot for transmission of a single cell or packet. The apparatus further comprises a processor for sequentially processing the entries on the timing ring to transmit particular packets or cells onto the network during their scheduled time slot or the first available time slot thereafter. When a dynamically allocated entry for a particular virtual circuit is processed, the next transmission of a packet on that virtual circuit is scheduled by dynamically allocating another entry on the timing ring. The timing ring may also comprise entries statically preallocated for the transmission of packets of other virtual circuits, typically CBR and real-time VBR circuits.

According to another aspect, the present invention is directed to a method for scheduling the transmission of cells or packets in an ATM, or other packet switching, network. The method comprises the steps of dynamically allocating entries on a timing ring for scheduling transmission of packets of variable rate virtual circuits, typically ABR, 5 UBR and non-real time VBR circuits. An entry on the timing ring represents an available time slot for transmission of a single packet. The method further comprises processing the entries on the timing ring sequentially to transmit particular packets onto the network during their scheduled time slot or the first available time slot thereafter, and then dynamically allocating new entries to schedule the transmission of the next packet for each virtual 10 circuit. The method may also comprise the step of preallocating entries on the timing ring for scheduling transmission of packets of fixed-rate virtual circuits, typically CBR or real-time VBR circuits.

The preferred embodiment of the present invention is essentially a hybrid of two related scheduling mechanisms: static and dynamic scheduling. The use of these two 15 mechanisms in combination permits trade-offs between flexibility and performance in the scheduling and transmission of cells and packets of various traffic classes. In this manner, packets of fixed-rate CBR and real-time VBR circuits are transmitted at fixed intervals as required by these connections, while packets of variable-rate ABR, VBR and UBR circuits are transmitted in compliance with their negotiated traffic contracts.

20

#### Brief Description of the Drawings

A more complete understanding of the invention may be obtained by reading the following description in conjunction with the appended drawings in which like elements are labeled similarly and in which:

25 Fig. 1 is a functional block diagram of an embodiment of a transmission pacing mechanism constructed in accordance with the principles of the present invention;

Fig. 2A is a diagram of an ATM cell as defined by the CCITT;

Fig. 2B is a diagram of an ATM cell header at the User-Network Interface as defined by the CCITT;

30 Fig. 2C is a diagram of an ATM cell header at the Network-Network Interface as defined by the CCITT;



Fig. 3 is an illustration of the timing ring employed in the pacing mechanism of Fig. 1;

Fig. 4 is a flow diagram of a timing ring processing algorithm employed in the pacing mechanism of Fig. 1;

5 Fig. 5 is a diagram of a static action data structure employed in the pacing mechanism of Fig. 1;

Fig. 6 is a diagram of a dynamic action data structure employed in the pacing mechanism of Fig. 1;

10 Fig. 7 is an example of the scheduling operation of the pacing mechanism of Fig. 1;

Fig. 8 is a diagram of a simplified state machine for AALS segmentation and transmission.

Fig. 9 is an illustration of a timing ring employed in a second embodiment of the pacing mechanism of Fig. 1;

15 Fig. 10 is an illustration of a timing ring employed in a third embodiment of the pacing mechanism of Fig. 1; and

Fig. 11 is an illustration of a timing ring employed in a fourth embodiment of the pacing mechanism of Fig. 1.

## 20 Detailed Description of the Invention

In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been  
25 shown in detail in order not to unnecessarily obscure the present invention.

Furthermore, although what is described herein is a transmission pacing mechanism for use in ATM networks, it should be understood that the present invention is in no way limited in applicability to ATM networks as defined by the CCITT. Rather, one skilled in the art will recognize that the invention described herein may be employed in a wide variety  
30 of packet switching networks. For examples of some alternative networks see de Prycker, pp. 50-58.

A block diagram of a pacing mechanism 100 constructed in accordance with the principles of the present invention is shown in Fig. 1. Pacing mechanism 100 comprises a microprocessor 120, DRAM memory 130, first-in-first-out (FIFO) transmit buffer 140, transmission controller 150, timer 160 and network line interface 170. Microprocessor 120, DRAM 130 and FIFO buffer 140 are interconnected via data bus 180.

Microprocessor 120 may be, for example, a commercially available RISC single-chip microprocessor such as the ARM610, manufactured by Advanced RISC Machines of Cambridge, England, operating at 32 MHz. Transmission controller 150 represents a block of control logic, the attributes and functionality of which will be described below. This logic may be implemented with discrete components, gate arrays, or Application Specific Integrated Circuits (ASICs) depending on the requirements of the particular implementation. The actual composition of network line interface 170, as will be recognized by the skilled artisan, depends upon the physical medium from which the ATM network is constructed. For example, UTP 802.4 Token Ring components may be employed for use with a 25.6 Mb/s ATM physical layer. Timer 160, DRAM 130 and FIFO buffer 140 may be selected from a wide variety of standard components known to those of ordinary skill in the art.

Various bits in the header of the ATM cell are used in carrying out the preferred embodiment of the following present invention. As an aid to understanding the invention, a description of a typical ATM cell is as follows. Fig. 2A shows the format of an ATM cell 200 as defined in CCITT Recommendation I.361. This format, adopted for use in B-ISDN and other wide-area networks, specifies a cell of 53 bytes (octets): an information field or payload 210 of 48 bytes (octets) which contains the user information which is the object of the transmission and a cell header 220 of 5 bytes (octets).

Cell header 220, or simply "header", is used for transmitting a variety of control information regarding the instant cell. Fig. 2B shows the structure of this header at the User Network Interface (UNI), that is the interface between an end-user device and an ATM switch. Here, the header is made up of a Generic Flow Control field 230 for specifying information which may be used to control traffic flow at the user-network interface, a VPI (virtual path identifier) 240, a VCI (virtual circuit identifier) 250, a payload type 260 which provides information regarding the type of information contained in the payload area 210 of the cell, a cell loss priority bit 270 for setting the priorities

relating to the abandonment of the cell during overload conditions, and a Header Error Control field 280 which contains an error control check byte for the previous four bytes (octets) in the header 220.

Fig. 2C shows the format of header 220 at the Network-to-Network Interface (NNI),  
5 the interface between network switches. This header structure is identical to the structure at the UNI except the Generic Flow Control (GFC) field 230 is replaced with four additional bits of VPI 240. ATM networks do not provide for flow control of the type which is implemented in some packet networks and ATM networks have no facility to store cells over a long period of time. Therefore, inside an ATM network there is no need for  
10 generic flow control. GFC 230 is defined in the UNI to provide header space for end-node processing of cells. GFC 230 can be used by an end-node for any purpose, and GFC 230 is employed by the present invention, as described below. Thus, within the NNI, GFC 230 may be eliminated in favor of an expanded VPI 240. For more information regarding standard ATM cell formats see de Prycker, pp. 124-28. One skilled in the art will  
15 recognize that the cell format described herein represents only one standard cell format and that alternative cell formats and sizes may be employed in conjunction with the present invention.

The preferred embodiment of the present invention is a hybrid implementation of two related scheduling mechanisms. The first, static cell scheduling, is primarily suited,  
20 although not constrained, to uniform rate cell pacing as required by CBR and real-time VBR traffic classes. The second, dynamic cell scheduling, is directed to variable rate cell pacing requirements, as defined for ABR, UBR and non-real-time VBR traffic classes. As will be discussed below in the context of alternate embodiments, either mechanism may be used in isolation. However, the use of the two mechanisms in combination permits  
25 appropriate trade-offs between flexibility and run-time performance for most network implementations.

The operation of the present invention centers around a "timing ring," a conceptual diagram of which is shown in Fig. 3. Although simple timing rings, also known as slot reservation controllers, have been employed previously in prior art scheduling mechanisms,  
30 the ring employed in the present invention is a sophisticated implementation designed to handle both static and dynamic scheduling simultaneously.

Timing ring 300 is shown in Fig. 3 as a linear table for convenience, but it is properly envisioned as circular. As discussed below, processing of END marker 350 will result in looping by the pacing algorithm to the front of timing ring 300. Each entry 305 on timing ring 300 represents an available cell transmission slot and corresponds to the transmission time for one ATM cell in the current network implementation. Entry 305 is composed of two fields: a static action 310 and a dynamic action 320. Together these fields define which "actions," normally cell transmissions, are to be performed during that time slot, and what actions may be performed subsequently. Those of ordinary skill in the art will recognize that although a single timing ring 300 is employed in the preferred embodiment of the present invention, a pair of parallel timing rings, one for static actions and another for dynamic actions, may also be used. Because both fields must be read for each time interval, however, a single ring implementation is more efficient.

Each action field 310, 320 physically contains either a NULL pointer or an address pointer to a data structure, described below, which is stored in DRAM 130 and which defines an action to be taken during that time interval. The NULL pointer indicates that the time slot has not been allocated for the particular scheduling mechanism, static or dynamic, and that no action need be performed. During a single time slot either the static or dynamic field, or both, may contain pointers to action data structures.

The static fields 310 are employed to pace the transmission of CBR and VBR virtual circuits which require transmission of cells at precise time intervals. These fields are preallocated for particular virtual circuits at connection set-up time, or when the quality of service parameters for a given virtual circuit are altered by the network. By preallocating time slots for a particular circuit at precise integral intervals, transmission of cells onto the network containing delay sensitive data can be achieved with a guaranteed maximum delay jitter.

For statically allocated transmissions, it will be recognized that the length of timing ring 300, that is, the number of time slots in the ring, is determined by the interface speed and the desired timing resolution. As the size of timing ring 300 increases, the timing resolution increases, but at the expense of memory needed to store timing ring 300. If the minimum service bandwidth (MSB) for the network implementation is known, the minimum size (N) in time slots of timing ring 300 may be calculated as follows

$$N = \frac{TMB}{MSB}$$

where TMB represents the overall total bandwidth of the transmission medium.

5 As shown in Fig. 3, only one static action 330 may be allocated for a single ring entry, and that action must occur during that corresponding time interval. In contrast, an allocated dynamic action field 320 contains a pointer to a queue of dynamic actions 340 representing actions which become *current*, that is, ready for processing, during that time slot. Because static actions always take precedence, a dynamic action allocated a particular  
10 ring entry might not be processed during that time slot. Therefore, the presence of a dynamic action at a particular ring entry indicates that the action may be processed immediately or as soon as possible afterwards.

Dynamic actions, suitable for cell transmission on ABR and UBR circuits, are not preallocated. Typically, only one dynamic action for a particular virtual circuit will be  
15 allocated on timing ring 300 at any one time. When that action is processed, the next cell transmission for that circuit will be scheduled by placing an action 340 on the dynamic action queue at the ring entry 305 corresponding to the earliest slot for transmission of that cell. This procedure will be described in more detail below. Because dynamic actions are scheduled in real-time, there is no intrinsic limit to the timing resolution for these actions.  
20 However, the length of timing ring 300 does determine the minimum rate at which cells may be transmitted.

Fig. 4 represents an overall flow diagram for the processing by microprocessor 120 of ring entries on timing ring 300. Processing of the timing ring 300 according to this algorithm is initiated in response to hardware timer interrupt 185, shown in Fig. 1. As  
25 microprocessor 120 can process and queue cells in FIFO buffer 140 much faster than they are physically transmitted onto the network, normally several entries are processed in a burst, rather than handling a single entry per actual time interval. This implementation reduces the fixed penalty overhead of entering and leaving the interrupt code. The maximum number of entries which may be processed per burst is limited by the maximum  
30 length of FIFO buffer 140.

Microprocessor 120 continually maintains a pointer to the "next" ring entry 305, that is, the next entry to be processed. After receiving interrupt 185, microprocessor 120 will

read both the static and dynamic fields of this current ring entry 305, as represented by block 410. Microprocessor 120 first checks the dynamic field 320, block 420, to determine if it contains a pointer to a queue of dynamic actions which have now become current. If so, microprocessor 120 moves to block 430, where that entire queue of dynamic actions is  
5 placed into another data structure known as the Latent Queue.

The Latent Queue is the mechanism by which all dynamic actions which have become current are queued for processing. As queues of dynamic actions are encountered in current ring entries, they are appended to the Latent Queue. The first action in the Latent Queue will be processed at the first ring entry for which there is no valid static  
10 action present, that is, when static field 310 is set to NULL. Dynamic actions are only processed from the Latent Queue; a dynamic action's presence on timing ring 300 simply indicates the moment at which that action may be moved to the Latent Queue and enabled for processing.

First-in, first-out (FIFO) list structures, as will be recognized by those of ordinary  
15 skill in the art, require special case handling for the first entry added to a list, and require both "head" and "tail" pointers to be maintained. With this additional overhead, manipulation of FIFO queues is a potentially slow operation for any processor. To avoid this overhead, the queues of dynamic actions maintained at each ring entry are singly-linked last-in, first-out (LIFO) queues, rather than FIFO queues. This arrangement requires lower  
20 overhead because all queue "add" and "remove" operations are performed at the queue head, requiring only a single pointer to be maintained.

The use of LIFO queues, however, results in local unfairness: queued actions are not processed in a strict "first come, first served" basis. To redress this, the Latent Queue is maintained as a FIFO stack which can guarantee long-term fairness. As entries on the  
25 ring are processed, any dynamic action LIFO queues that are encountered are pushed onto the FIFO Latent Queue exactly as found. In other words, the FIFO Latent Queue contains LIFO queues of actions. Of course, other implementations of these queues are possible, but this arrangement yields a minimum of processor overhead.

Regardless of whether or not a dynamic action queue was encountered and pushed  
30 onto the Latent Queue, microprocessor 120 will determine if a static action is present for that ring entry, as indicated in block 440. This is actually a two-part test. First, if static

field 310 contains a NULL pointer, then no static action is scheduled for that time slot. However, it is also possible that a static action is scheduled for that time slot. *e.g.*, the transmission of a cell on a particular CBR circuit, but that the application program employing that circuit has failed to supply the requisite data for transmission. Therefore,  
5 a check can be made to see if data is present. If no data is waiting, then the slot may be used for the processing of a dynamic action as if the slot had not been preallocated.

If the ring entry is statically allocated, and if data is present to transmit, then microprocessor 120 steps to block 460 to read the action data structure indicated by the pointer in static action field 310. A typical static action data structure 500 is shown in Fig.  
10 5. Microprocessor 120 will make a subroutine call to an action handler routine located in memory at the address specified by handler pointer 550. This cell transmission handler routine will create a single ATM cell by concatenating the four bytes (octets) of cell header 540 with 48 bytes (octets) of payload data (if a standard cell size is being employed) which is located in memory at the address specified by data pointer 520. The handler routine will  
15 then cause this ATM cell to be placed at the back of the transmit queue in FIFO buffer 140 to await transmission by transmission controller 150. The handler routine will then complete any other specified tasks and execute a subroutine return instruction to return control to the flow diagram of Fig. 4.

If microprocessor 120 determines at block 440 that static field 310 is not allocated,  
20 or that data is not waiting for an allocated field, microprocessor 120 will proceed to check the status of the Latent Queue, as represented in block 450. If at least one dynamic action is resident in the Latent Queue, microprocessor 120 will move to block 470 to read that action off the head of the Latent Queue.

A typical dynamic action data structure 600 is shown in Fig. 6. The first five fields in  
25 dynamic action data structure 600 are identical to those contained in static action data structure 500. The additional fields shown in Fig. 6 are employed in scheduling subsequent dynamic actions for a particular virtual circuit and are described more fully below. As with static actions, microprocessor 120 will make a subroutine call to a action handler routine located in memory at the address specified by handler pointer 550. Again, this cell  
30 transmission handler routine will create a single ATM cell by concatenating the four bytes (octets) of cell header 540 with 48 bytes (octets) of payload data (if a standard cell size is

being employed) which is located in memory at the address specified by data pointer 520. As before, this ATM cell will be placed at the back of the transmit queue in FIFO buffer 140 to await transmission by transmission controller 150. The handler routine will then schedule the next cell transmission for this virtual circuit by appending a dynamic action  
5 on timing ring 300 at the time slot representing the earliest time the next cell should be transmitted. The calculation of this interval will be described more fully below. The handler routine then completes any other specified tasks and executes a subroutine return instruction to return control to the flow diagram of Fig. 4.

If there is no static action to be implemented, and the Latent Queue is empty,  
10 microprocessor 120 will move from block 450 to block 480. This condition represents a time slot where no cell transmission should take place. However, since ring entries are processed in a burst, simply moving to next allocated ring entry and placing that cell in the transmit buffer would result in that next cell being transmitted too early. The simplest solution would be to construct a dummy ATM cell with VCI 250 set to zero and place that  
15 cell in FIFO buffer 140. This cell would be transmitted onto the network in due course by transmission controller 150. Its presence would occupy that transmission slot, thus ensuring that subsequent cells would be transmitted in their scheduled slots. Such use of dummy cells is permissible, because an ATM network will automatically discard any cell containing a VCI 250 set to zero. However, this results in additional processing within the network  
20 which should be avoided if possible. Another, more efficient, mechanism is to employ an intelligent transmission controller 150 which can delay the transmission of a cell until its appointed transmission slot. This mechanism is implemented in the present invention by encoding a cell delay count in the Generic Flow Control (GFC) field 230 of each outgoing cell header. The value encoded in this field indicates to transmission controller 150 the  
25 number of cell times to wait before beginning transmission of that cell. Transmission controller 150 accomplishes this delay by monitoring cell delay timer 190, a timer signal which fires once per time slot.

Thus, after encountering an empty ring entry, microprocessor will increment a delay counter in block 480. This delay counter is always maintained and its current value is  
30 placed in GFC 230 of all constructed outgoing cells by the action handler routine. The



action handler routine also resets the delay counter to zero when it places a newly-constructed cell into FIFO buffer 140 to await transmission.

Of course, GFC 230 consists of only four bits; thus, cell delay times between zero and fifteen (15) cell transmission times are possible. If microprocessor 120 increments the  
5 delay counter to sixteen (16) in block 480, it will reset the delay counter to zero, construct a dummy ATM cell, as described above, and place that cell in FIFO buffer 140 to await transmission.

Regardless of what tasks were performed in processing the current ring entry, microprocessor 120 will ultimately proceed to block 490 where it will advance to the next  
10 ring entry and loop around for processing. The actual physical incrementing of the pointer to the next ring entry may occur anywhere in the algorithm loop; in fact, it may be more efficient to advance the pointer when reading the entry at block 410. This step is shown at the bottom of the flow diagram of Fig. 4 for conceptual clarity.

The loop in the flow diagram of Fig. 4 is shown as perpetual for the sake of  
15 convenience. However, the processing will break out of this loop after it has processed a preset number of ring entries (or, possibly, loaded a preset number of cells into FIFO buffer 140). After breaking out, the processing will wait until it again receives timer interrupt 185 to resume processing ring entries. The preset number of ring entries and the frequency of timer interrupt 185 are implementation dependent, but these values should be  
20 chosen so that the transmission hardware is never left idle. In other words, FIFO buffer 140 should not be permitted to empty; rather the processing should be "topping off" ("topping up") the set of cells that the transmission hardware is trying to transmit. In this fashion, transmission controller 150 will always be transmitting cells or implementing cell transmission delays.

25 Fig. 7 presents an example of the scheduling mechanisms in operation, showing a window of ten time slots,  $T_1$  to  $T_{10}$ . Timeline 710 shows which actions are performed in which time slots given the allocated timing ring 705. In time slot  $T_1$  static action  $S_1$  is performed and dynamic actions  $D_1$ ,  $D_2$  and  $D_3$  are placed on the Latent Queue. Time slots  $T_2$  and  $T_3$  are unallocated, so the top two actions on the Latent Queue,  $D_1$  and  $D_2$ , are  
30 performed in those time slots, respectively. Slots  $T_4$ ,  $T_5$  and  $T_6$  are allocated for static actions  $S_2$ ,  $S_3$  and  $S_4$ , respectively, and these actions are performed in these time slots,

while the dynamic queues allocated to slots  $T_5$  and  $T_6$  are placed on the Latent Queue. The top four actions on the Latent Queue,  $D_3$  through  $D_6$ , are performed in slots  $T_7$  through  $T_{10}$ . Note that although action  $D_9$  is allocated to slot  $T_8$ , it is not performed in that time slot, but rather placed on the bottom of the Latent Queue. After slot  $T_{10}$ , actions  $D_7$ ,  $D_8$  and  $D_9$  remain on the Latent Queue and will be performed subsequently.

As mentioned above, dynamic actions, used for ABR and UBR cell transmission, are allocated in real-time as cells from these virtual circuits are processed. When a dynamic action is handled, the action handler routine is charged with the task of scheduling the next dynamic action for that particular virtual circuit. The additional fields in the dynamic action data structure 600 enable the handler routine to perform this scheduling.

Scheduling information is stored as integral plus fractional time periods. The integral transmit period is stored in units of cell transmission times in Integral Tx Period 610, whereas the fractional period is stored as number of transmitted cells in Counter 2 Rate 640. The scheduling operates as follows: the action handler will place the next dynamic action at timing ring entry 305 X entries from the current ring entry, where X is defined by Integral Tx Period 610. However, every Y number of transmitted cells on that particular virtual circuit, where Y is defined by Counter 2 Rate 640, the dynamic action will either be placed at one additional or one less ring entry (depending on the implementation) from the one defined by Integral Tx Period 610. Therefore, fine grained control can be achieved over the transmission rate. Counter 2 Value 660 maintains the count of cells transmitted on that virtual circuit for comparison to Counter 2 Rate 640. This value is reset when the fractional transmit correction is performed.

Counter 1 Rate 630 defines the number of data cells to be transmitted on a particular ABR virtual circuit before an ABR Resource Management (RM) cell is transmitted. Counter 1 Value 650 maintains the actual cell count. When Counter 1 Value 650 reaches the value in Counter 1 Rate 630, the action handler will transmit an ABR RM cell rather than a data cell. In the preferred embodiment of the present invention, UBR virtual circuits are handled the same way as ABR circuits, but Counter 1 Rate 630 is set to zero so no RM cells are generated.

For ABR scheduling, the current transmission rate is also represented by Rate ID 620. Rate ID 620 is a sixteen (16) bit number which is linearly mapped over a range of

rates which lie between one (1) cell per second and the line rate. As is known to those of ordinary skill in the art, the ATM Forum ABR specification requires both linear and exponential rate changes. These changes are handled by applying the linear or exponential rate change to Rate ID 620, and then using the newly-calculated value as an index to a  
5 look-up table to transform the result into an integral and fractional time period stored, respectively, in Integral TX Period 610 and Counter 2 Rate 640. For efficiency, ABR rate changes are normally applied only when an RM cell is transmitted, typically every thirty-two (32) data cells.

It should be recognized that the dynamic action data structure 600 may also be used  
10 for static actions. The counter fields may be used to implement transmit "gating", such that every 'N' of 256 allocated cell transmission slots are ignored. In other words, the static action handler will not transmit a cell in that slot. In this manner, fine-grained control over CBR rates, and implementation of VBR rate variability may be achieved.

One of ordinary skill in the art will recognize that actions, together with their  
15 corresponding action handler routines, may be used to implement simple state machines. In fact, such a state machine is used to implement the integral plus fractional time period scheduling. As a further example, a simplified state machine for AAL5 segmentation and transmission (for multiple cell AAL5 packets) is shown graphically in Fig. 8. Once an AAL5 packet is presented by the application code for transmission, all fields in that virtual  
20 circuit's action data structure 500 (top half of structure 600) are initialized. In state 1 810, the action handler routine will initialize the AAL5 CRC calculation, send the first cell in the AAL5 packet, save the partial CRC residue in CRC 510 in the action data structure, schedule the next transmission if it is a dynamically scheduled circuit, and switch to the next state. When the next action is performed for that virtual circuit, the action handler  
25 routine will be in state 2 820. Here the handler routine will restore the CRC residue, send the next cell in the AAL5 packet, save the partial CRC residue in CRC 510, schedule the next transmission, and switch to the next state if the next cell is the last cell in the packet. This will be indicated when the updated data pointer 520 is equal to the value in last cell pointer 530. The action handler will remain in state 2 820 until this last cell is reached.  
30 In state 3 830, the action handler routine will restore the CRC residue, send the last cell in the AAL5 packet with the final computed CRC value, and return a transmitted completed

status code to the application program. This type of state machine, as will be apparent to those of ordinary skill in the art, can be implemented for a wide variety of network operations.

It is also important to note that actions are not constrained to performing cell transmission: rather, virtually any processing function that needs to be scheduled for a particular moment in time may be implemented in an action handler routine and specified by an action pointer on timing ring 300. In fact, special "static" actions are predefined for software management tasks. For example, an action may be placed in timing ring 300 every fixed number of slots which causes microprocessor 120 to break out of the loop defined by the flow diagram of Fig. 4 and return from interrupt processing. In addition, the END marker 350 may be defined as an action which results in pointer address wrap-around to the start of timing ring 300. Nor do these management actions need to consume a physical time slot. Microprocessor 120 may simply move to the next ring entry 305 for processing any cells for transmission in that physical time slot.

Another possible use for a timing ring slot is the allocation of an action to return a "receive" buffer to an application program. Standard non-AAL5 traffic does not contain any end-of-packet marker; consequently, each received cell is typically passed directly to the application upon reception. This practice, however, incurs substantial processing overhead. Improved efficiency can be achieved by buffering several received cells and then passing those cells as a single block to the application.

However, because no packet structure can be inferred from the received traffic stream, a few cells could languish in a partially filled receive buffer. The timing ring can be used to schedule a action which guarantees return of the receive buffer at timely intervals, irrespective of the number of cells currently stored in the buffer. This mechanism provides an essential efficiency enhancement, while guaranteeing a maximum receive latency.

The above preferred embodiment of the present invention is a hybrid implementation, combining dynamic and static scheduling. This arrangement will provide appropriate tradeoffs between efficiency and flexibility for most network schemes. However, other embodiments are possible, and may be preferred for special network applications.

A second embodiment consists of a timing ring 300 implementing static action handling only, as shown in Fig. 9. True static actions, such as CBR and real-time VBR cell transmissions, are statically preallocated at connection set-up time as above in the first embodiment. ABR cell transmissions may also be implemented by using a form of "pseudo-static" scheduling, that is, modifying the contents of the static fields 310 of timing ring 300 in real time. This occurs in the following manner: as shown in Fig. 9 a dynamic action is to be performed at time slot  $T_2$ . As this action is performed, the next action for that virtual circuit must be scheduled because it has not been statically preallocated. Assume that Integral Tx Period 610 indicates that this next action should ideally be scheduled for time slot  $T_5$ . However, because that time slot is already preallocated to static action  $S_3$ , the processor must linearly search from  $T_5$  through timing ring 300 until a "free" time slot is found, here time slot  $T_7$ . The next dynamic action will be allocated to that slot.

This embodiment is suitable for network implementations consisting of many CBR and real-time VBR connections, but few simultaneous ABR transmitters. As the number of ABR transmitters increases, the length of the average ring search for scheduling increases, eventually making the processing overhead prohibitively expensive.

A third embodiment consists of a timing ring 300 implementing dynamic scheduling only, as shown in Fig. 10. This scenario operates precisely as dynamic scheduling in the above hybrid implementation, though there are no statically preallocated time slots. At each ring entry, dynamic action queues are placed onto the Latent Queue and the action at the top of the Latent Queue is performed. This arrangement is preferred for network implementations consisting mainly of ABR, UBR and non-real time VBR traffic circuits.

A fourth embodiment consists of a timing ring 300 implementing dual dynamic scheduling, as shown in Fig. 11. The upper fields 1110 represent high-priority dynamic scheduling, whereas the lower fields 1120 represent low-priority dynamic scheduling. In this implementation, two Latent Queues are employed, one for high-priority traffic, another for low-priority traffic. All actions resident on the high-priority Latent Queue are performed before actions on the low-priority Latent Queue. This arrangement provides for better service guarantees than the single dynamic action handling embodiment, but performance will degrade with a large number of simultaneous transmitters.

Although the above description is with respect to the use of the present invention as a transmission pacing mechanism at end-points of an ATM network, those of ordinary skill in the art will understand that the present invention may also be employed at internal nodes, or switching nodes, of an ATM network. Such an application may be desirable to  
5 perform traffic shaping to correct pacing deviations created by idiosyncracies within the network itself.

It should be understood that various other modifications will also be readily apparent to those skilled in the art without departing from the scope and spirit of the invention. Accordingly, it is not intended that the scope of the claims appended hereto be limited to  
10 the description set forth herein, but rather that the claims be construed as encompassing all the features of the patentable novelty that reside in the present invention, including all features that would be treated as equivalents thereof by those skilled in the art to which this invention pertains.

15

20

25

30

**CLAIMS:**

1. A method for scheduling the transmission of packets in a packet switching network comprising the steps of:  
dynamically allocating entries on a timing ring for scheduling transmission  
5 of packets of one or more virtual circuits in which an entry on the timing ring represents an available time slot for transmission of a single packet; and  
processing the entries on the timing ring sequentially to transmit packets onto the network during their scheduled time slots or the first available time slot thereafter.
- 10 2. The method according to claim 1 further comprising the step of preallocating entries on the timing ring for scheduling transmission of packets of one or more additional virtual circuits.
3. The method according to claim 2 wherein the processing step comprises  
15 placing packets into a transmit buffer to await transmission onto the network.
4. The method according to claim 3 wherein the dynamically allocating step occurs initially at virtual circuit establishment and thereafter when a packet scheduled by a dynamically allocated entry is placed into the transmit buffer.  
20
5. The method according to claim 4 further comprising the step of delaying the transmission of a packet onto the network for a number of cell transmission times equivalent to the number of previous ring entries for which no packets were placed into the transmit buffer.  
25
6. A packet transmission apparatus for scheduling the transmission of packets onto a packet switching network comprising:  
a timing ring containing entries dynamically allocated for the scheduling of the transmission of packets of one or more virtual circuits, in which an entry on the timing  
30 ring represents an available time slot for transmission of a single packet; and

a processor for sequentially processing the entries on the timing ring to transmit packets onto the network during their scheduled time slot or the first available time slot thereafter.

5           7.     The packet transmission apparatus of claim 6 wherein the timing ring includes entries preallocated for the scheduling of the transmission of packets of one or more additional virtual circuits.

8.     The packet transmission apparatus of claim 7 wherein the packet switching  
10 network is an ATM network.

9.     The packet transmission apparatus of claim 8 wherein the preallocated entries on the timing ring correspond to CBR or real-time VBR virtual circuits.

15          10.    The packet transmission apparatus of claim 9 wherein the preallocated entries on the timing ring corresponding to a single virtual circuit are spaced at fixed integral intervals.

11.    The packet transmission apparatus of claim 8 wherein the dynamically  
20 allocated entries on the timing ring correspond to ABR, UBR and non-realtime VBR virtual circuits.

12.    The packet transmission apparatus of claim 11 wherein only one dynamically  
allocated entry per virtual circuit is present on the timing ring at any one time.

25

13.    The packet transmission apparatus of claim 7 wherein an entry is dynamically allocated for a virtual circuit initially at connection set-up time and subsequently after a previously dynamically allocated entry for that virtual circuit has been processed.

30



14. The packet transmission apparatus of claim 13 wherein the next entry to be dynamically allocated for a virtual circuit is a fixed integral number of entries following the currently processed entry.

5 15. The packet transmission apparatus of claim 14 wherein the fixed integral number is modified by adding or subtracting one entry after a fixed number of packets on that virtual circuit have been transmitted.

16. The packet transmission apparatus of claim 7 wherein an entry on the timing  
10 ring may be both preallocated and dynamically allocated.

17. The packet transmission apparatus of claim 16 wherein an entry on the timing ring comprises a static action field and a dynamic action field.

15 18. The packet transmission apparatus of claim 17 wherein a preallocated entry is represented by an action data structure specified by the corresponding static action field, and wherein a dynamically allocated entry is represented by a queue of action data structures specified by the corresponding dynamic action field.

20 19. The packet transmission apparatus of claim 18 wherein the processor processes an allocated ring entry by placing the queue of action data structures onto a latent queue, and if preallocated, by performing the action indicated by the action data structure specified by the static action field.

25 20. The packet transmission apparatus of claim 19 wherein the processor processes a ring entry which has not been preallocated by performing the action indicated by the action data structure at the head of the latent queue.

30 21. The packet transmission apparatus of claim 20 wherein the action is performed by an action handler routine specified by the action data structure.

22. The packet transmission apparatus of claim 21 wherein the action performed includes the placing of a packet corresponding to a virtual circuit into a transmit buffer to await subsequent transmission onto the network.

5 23. The packet transmission apparatus of claim 22 wherein the transmission of the packet is delayed by a number of packet transmission times equivalent to the number of previous ring entries for which no packet transmission was specified.

24. The packet transmission apparatus of claim 23 wherein the number of packet  
10 transmission times is encoded in the header of the packet to be transmitted.

25. The packet transmission apparatus of claim 21 wherein the action handler routine implements a state machine such that state changes occur as subsequent actions corresponding to a single virtual circuit are performed.  
15

26. The packet transmission apparatus of claim 25 wherein the state machine implements AAL5 segmentation and transmission.

27. The packet transmission apparatus of claim 13 wherein the next entry to be  
20 dynamically allocated for a virtual circuit is determined by moving a fixed integral number of entries following the currently processed entry and then linearly searching from that entry to find the first freely available entry.

28. The packet transmission apparatus of claim 6 wherein an entry on the timing  
25 ring comprises a dynamic action field.

29. The packet transmission apparatus of claim 28 wherein a dynamically allocated entry is represented by a queue of action data structures specified by the corresponding dynamic action field.  
30

30. The packet transmission apparatus of claim 18 wherein the processor processes an allocated ring entry by placing the queue of action data structures onto a latent queue and by performing the action indicated by the action data structure at the head of the latent queue.

5

10

15

20

25

30

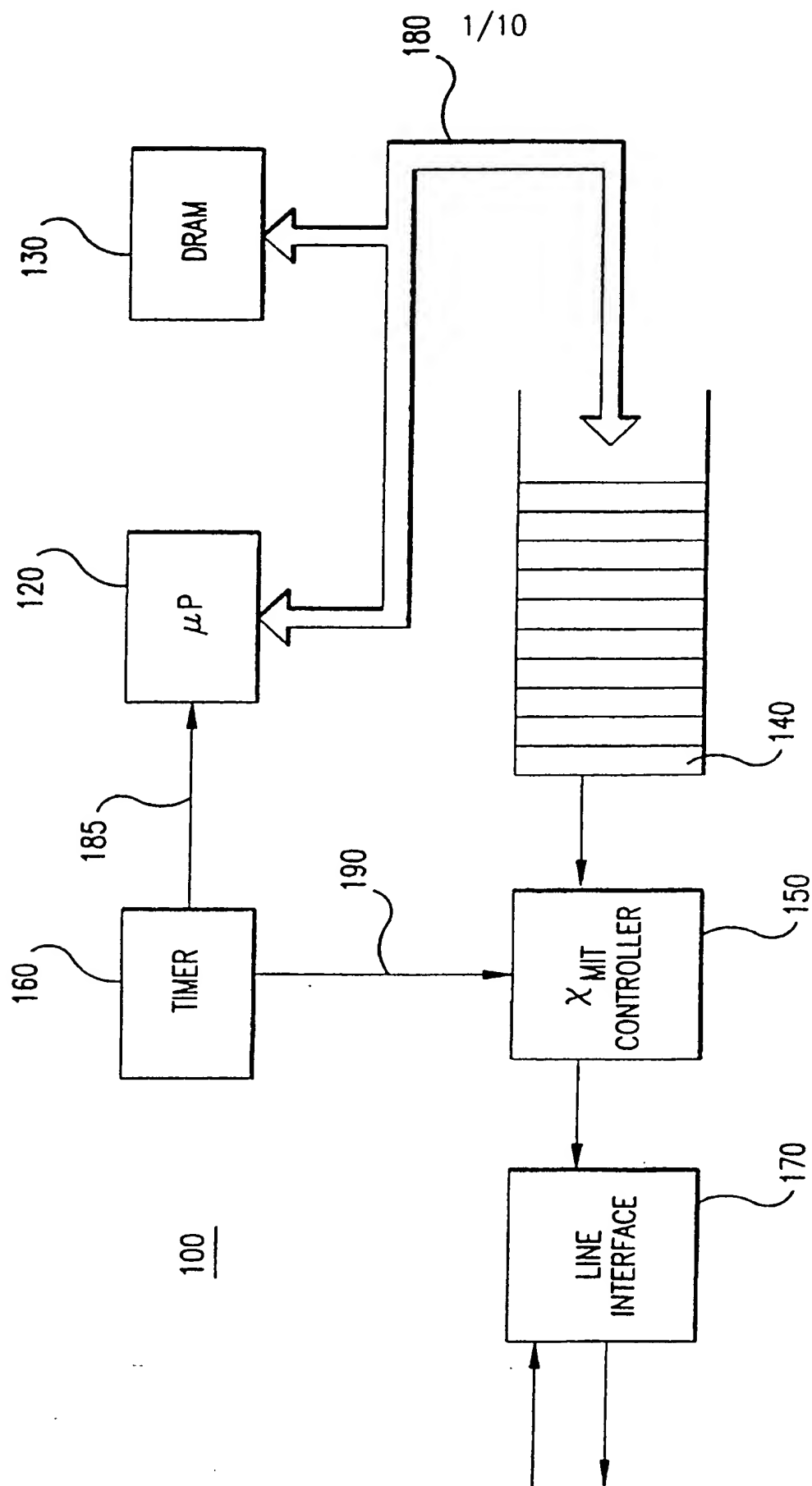


FIG.1

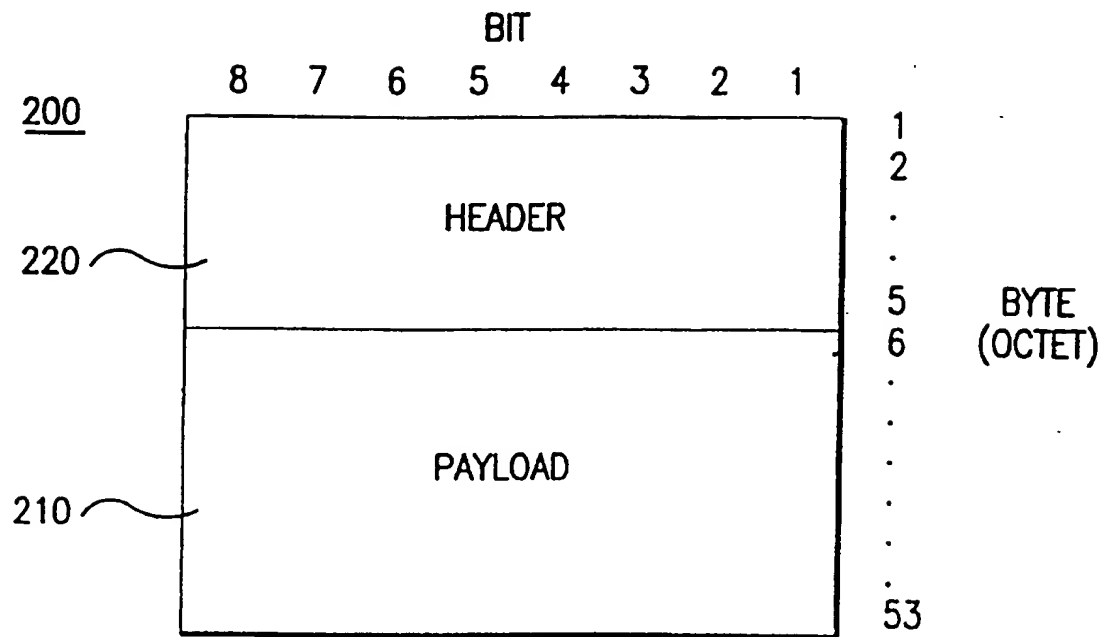


FIG.2A

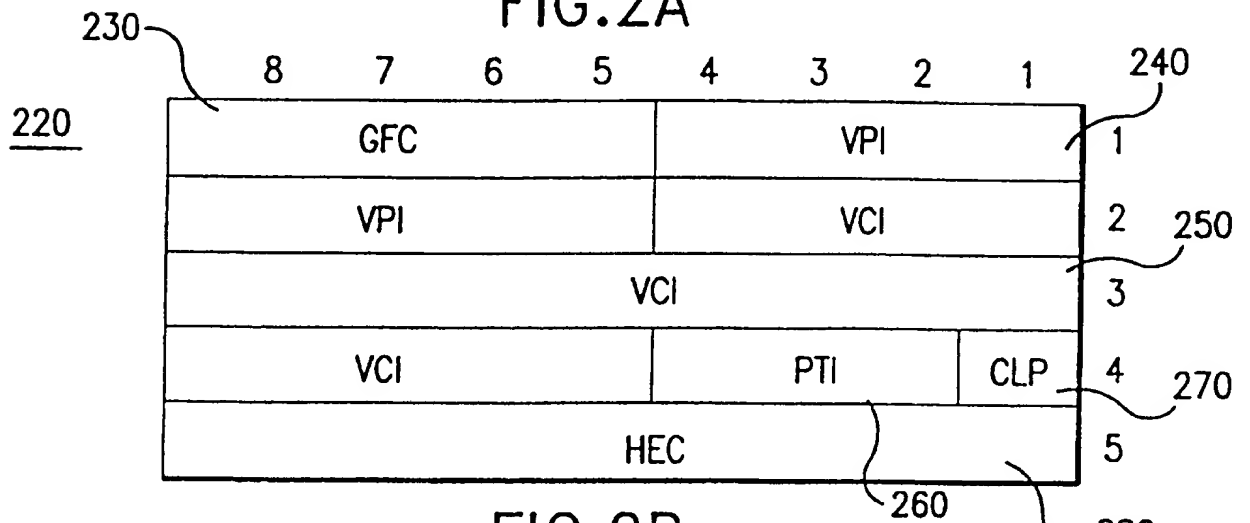


FIG.2B

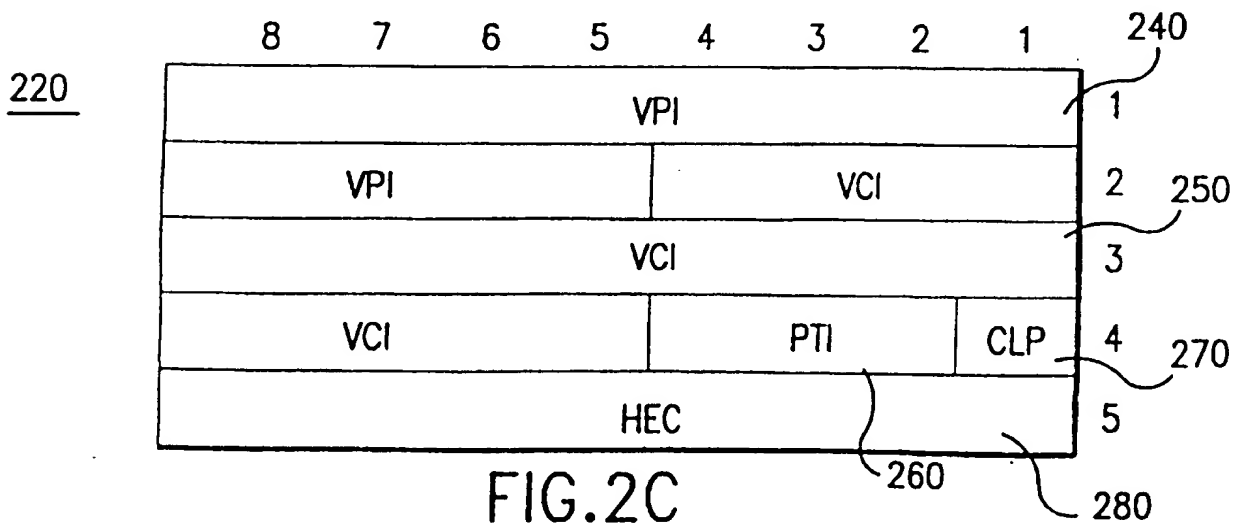
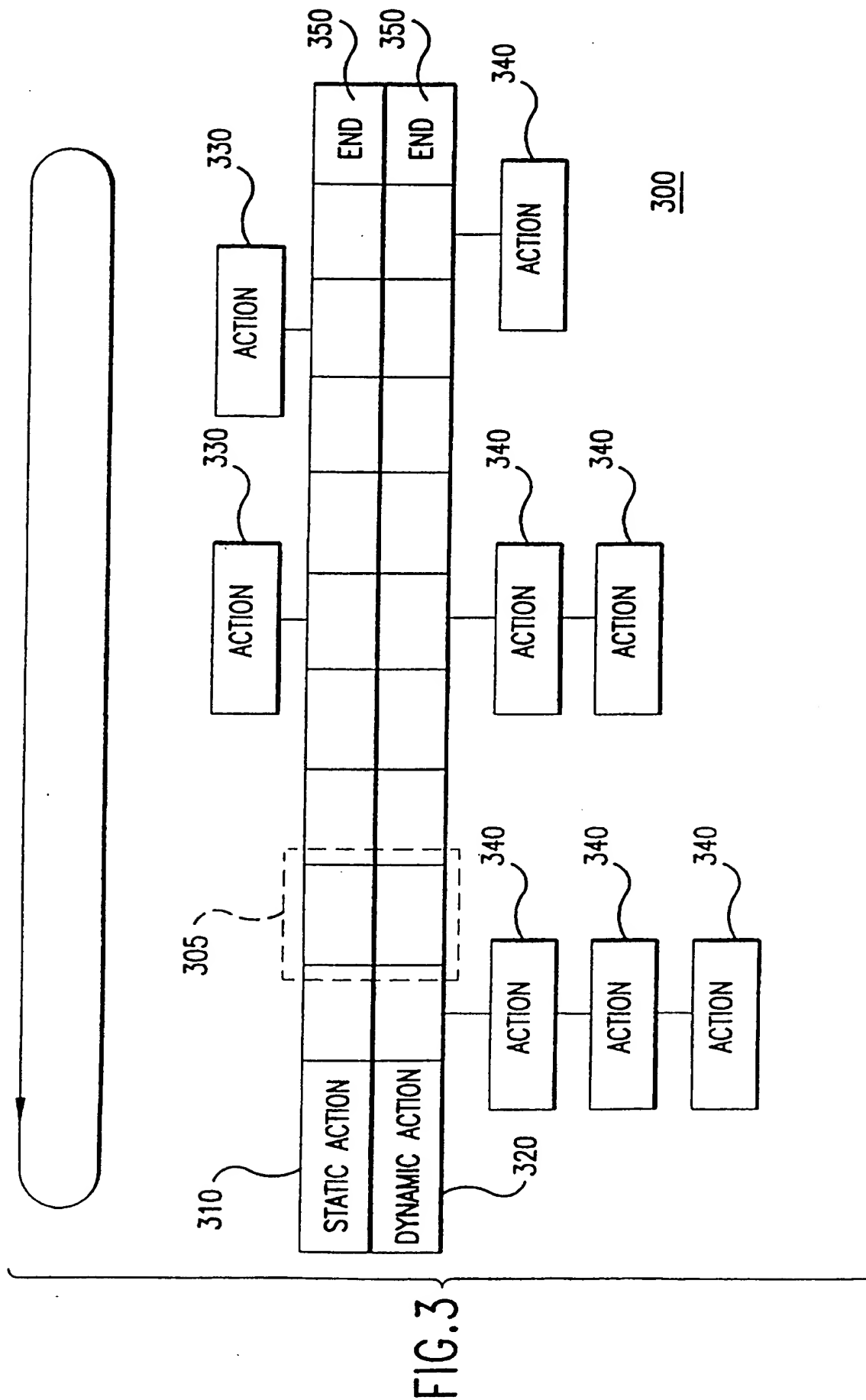


FIG.2C



4/10

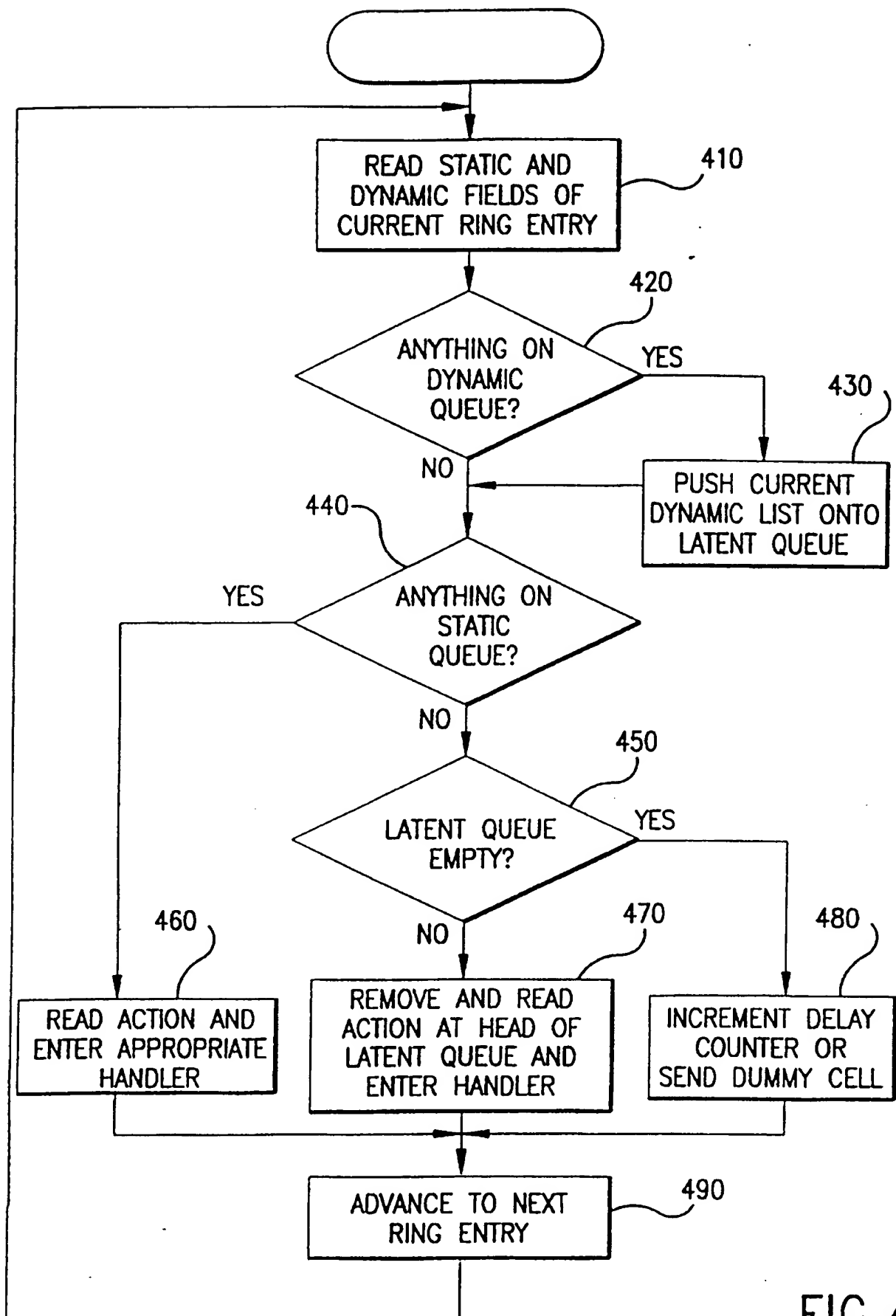
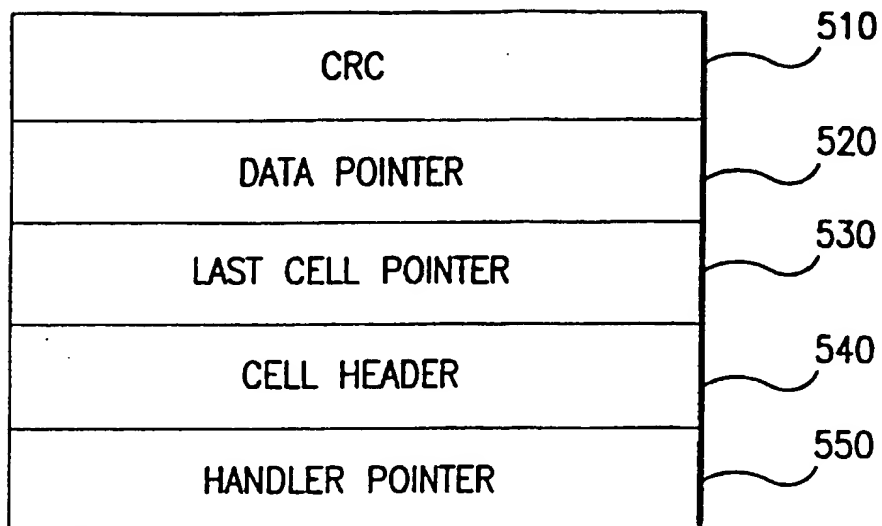
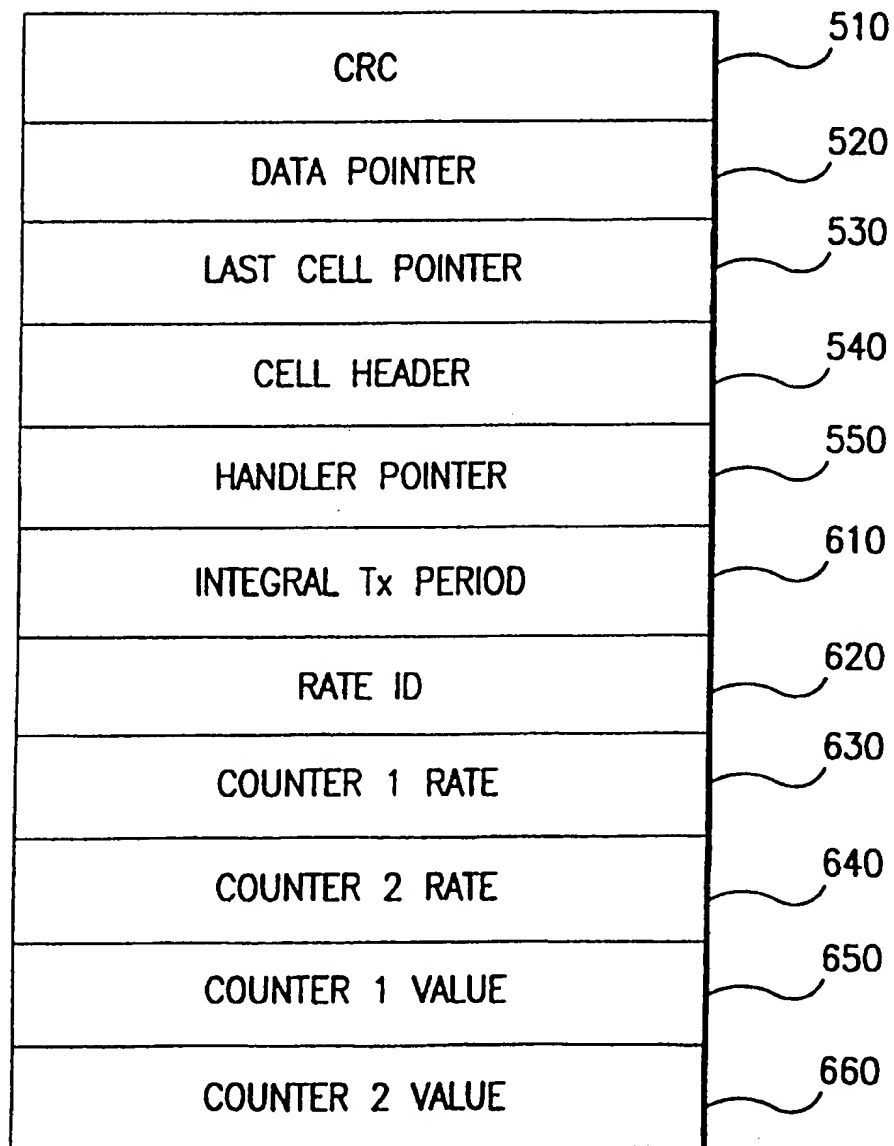


FIG. 4

500  
FIG.5

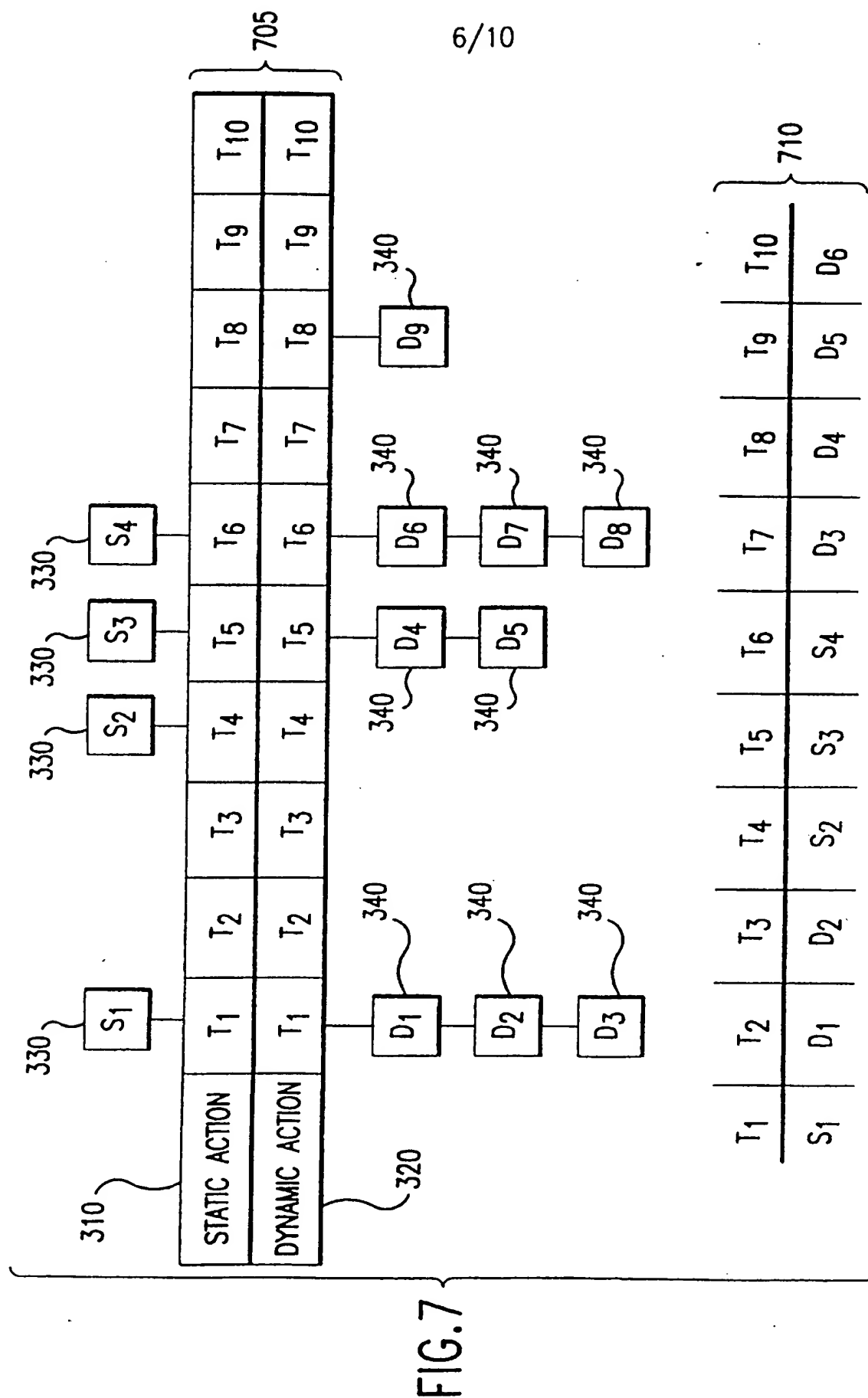


600  
FIG.6





6/10



7/10

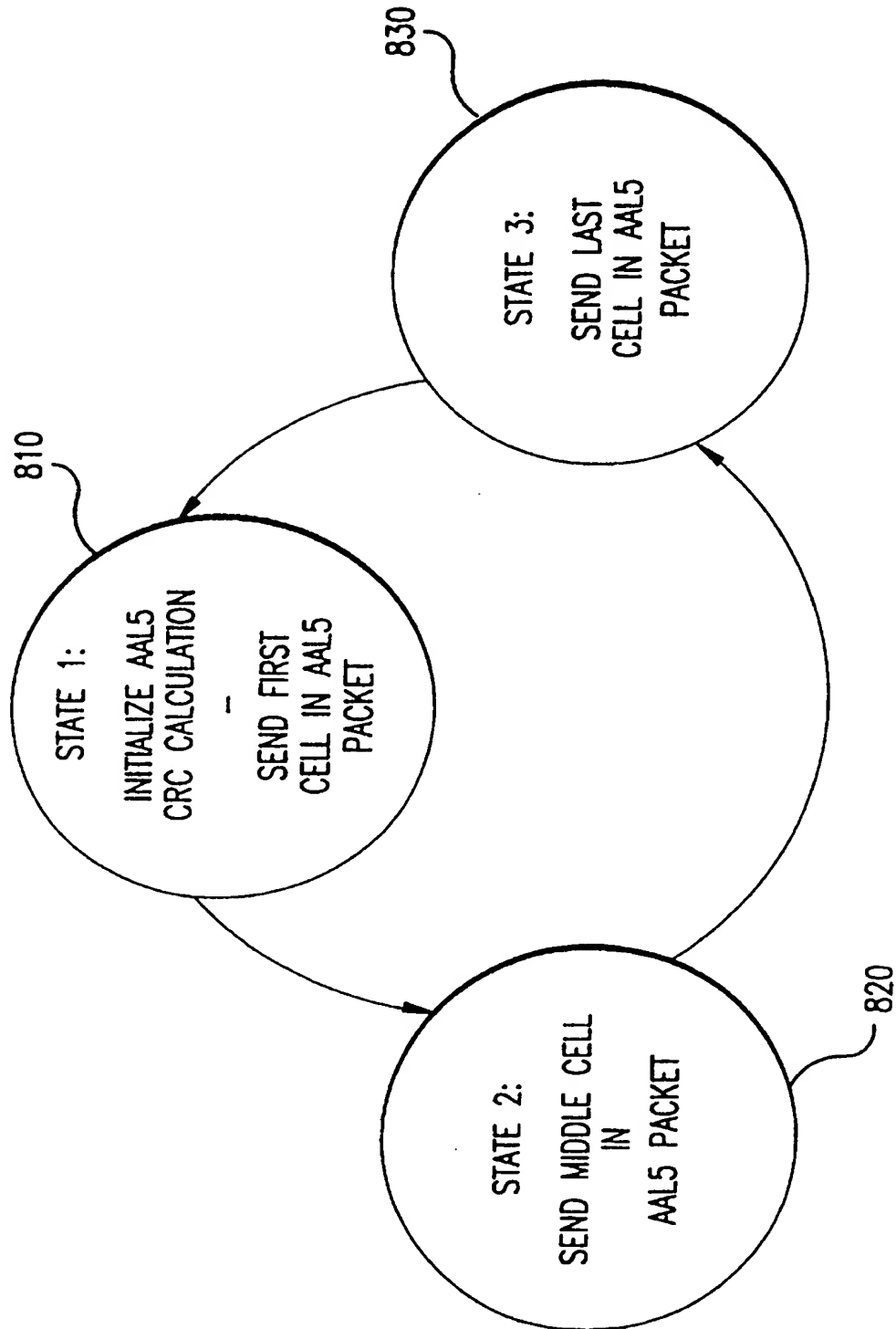


FIG.8

8/10

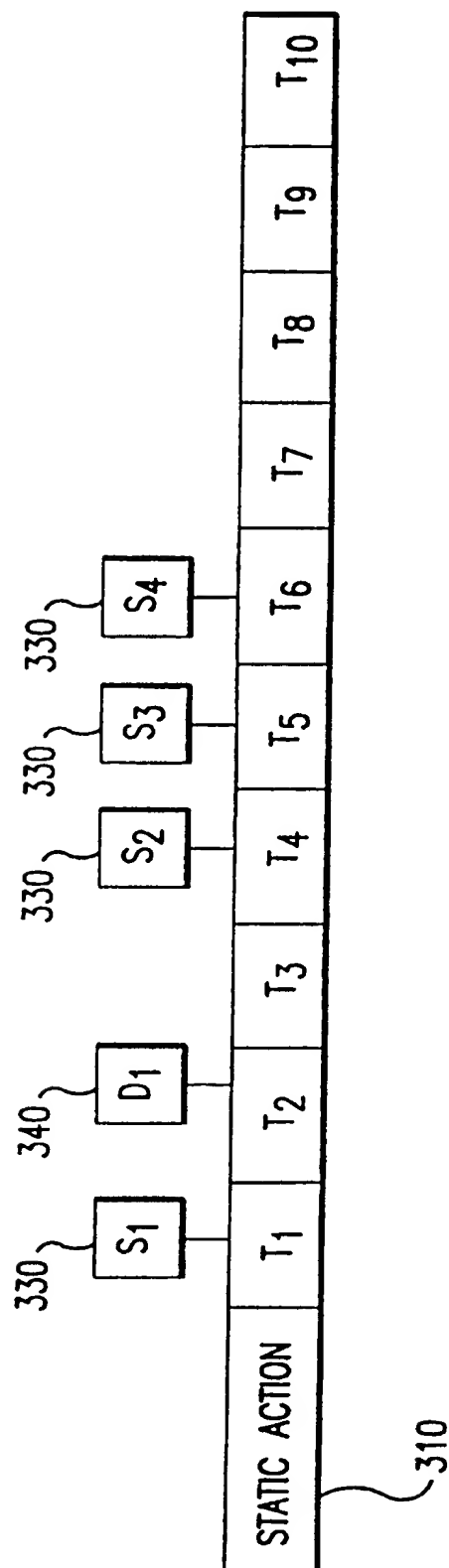


FIG.9

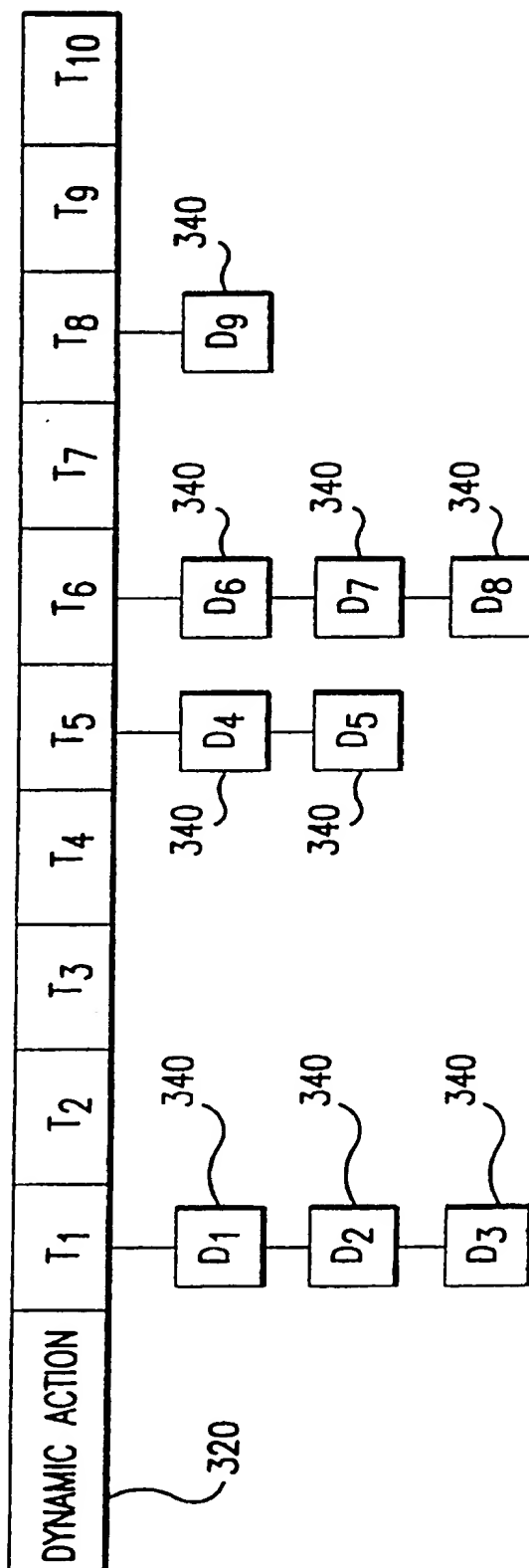


FIG.10

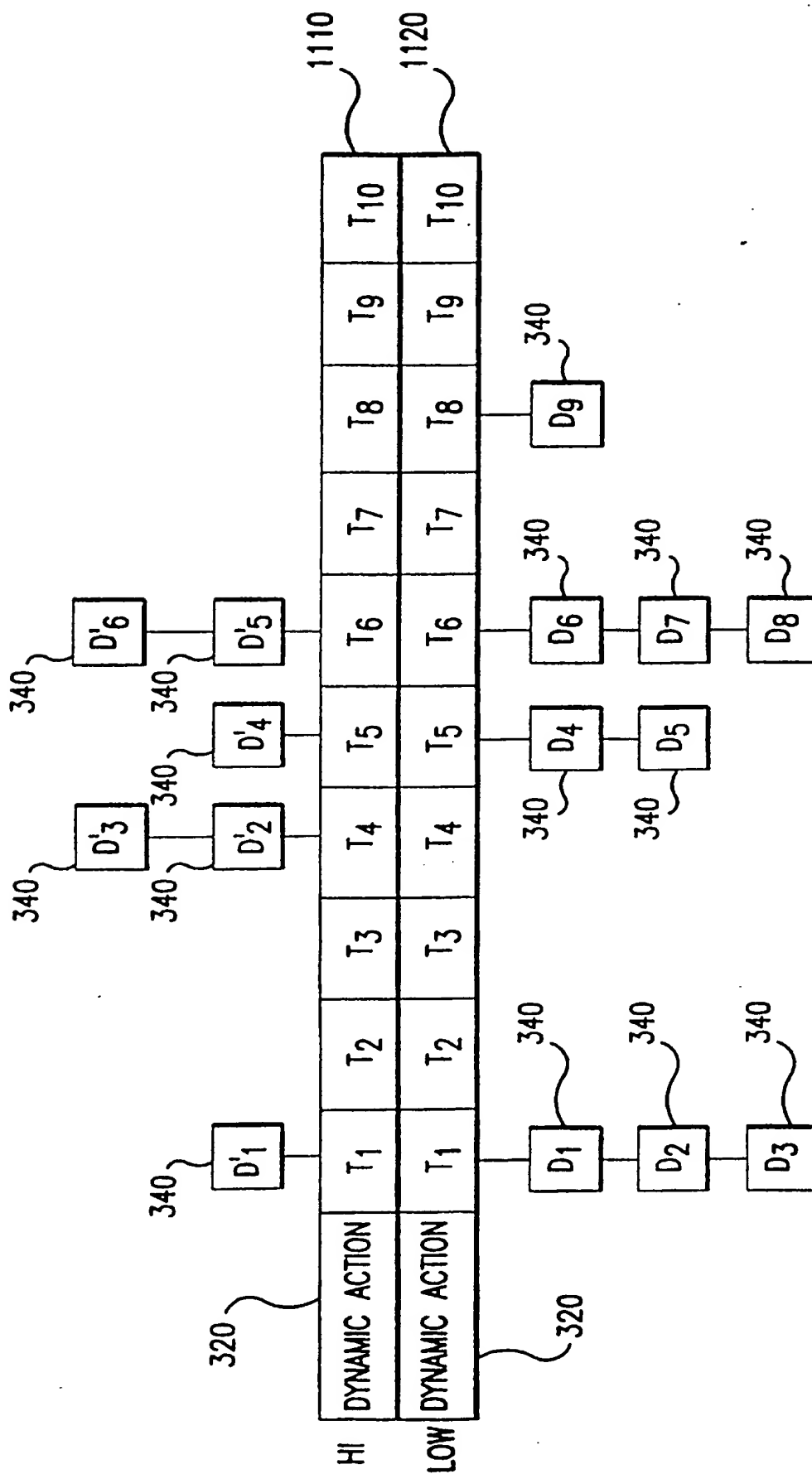


FIG. 11

# INTERNATIONAL SEARCH REPORT

Intern. Application No  
PCT/GB 97/02118

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 6 H04L12/56 H04Q11/04

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 596 624 A (AMERICAN TELEPHONE & TELEGRAPH) 11 May 1994	1-3,6-9, 11
Y	see column 1, line 55 - column 2, line 37	4,10,14, 16
	see column 5, line 57 - column 6, line 46	
	---	
Y	EP 0 618 749 A (TRT TELECOM RADIO ELECTR ; PHILIPS NV (NL)) 5 October 1994	10,14
	see column 3, line 15 - line 28	
	see column 3, line 45 - column 4, line 5	
	see column 4, line 36 - column 5, line 7	
	---	
	-/-	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

### \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- \*&\* document member of the same patent family

Date of the actual completion of the international search

7 November 1997

Date of mailing of the international search report

18. 11. 97

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 851 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Veen, G

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/GB 97/02118

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>CHIPALKATTI R ET AL: "SCHEDULING POLICIES FOR REAL-TIME AND NON-REAL-TIME TRAFFIC IN A STATISTICAL MULTIPLEXER"</p> <p>TECHNOLOGY: EMERGING OR CONVERGING?, OTTAWA, APRIL 23 - 27, 1989, vol. 3, 23 April 1989, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 774-783, XP000671034</p> <p>see page 775, right-hand column, line 7 - line 48</p> <p>see page 777, right-hand column, line 50 - page 778, left-hand column, line 8</p> <p>---</p>	4,16
A	<p>MERCANKOSK G ET AL: "MULTIPLEXING SPACER OUTPUTS ON CELL EMISSIONS"</p> <p>PROCEEDINGS OF INFOCOM '95 - CONFERENCE ON COMPUTER COMMUNICATIONS, FOURTEENTH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES, BOSTON APR. 2 - 6, 1995, vol. 3, 2 April 1995, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 49-55, XP000580562</p> <p>see page 49, right-hand column, line 6 - line 8</p> <p>see page 50, left-hand column, line 15 - page 52, right-hand column, line 15</p> <p>-----</p>	1-3, 6-11,14

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 97/02118

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0596624 A	11-05-94	CA 2104753 A JP 7079232 A US 5463620 A	30-04-94 20-03-95 31-10-95
EP 0618749 A	05-10-94	FR 2703540 A JP 7030541 A US 5594723 A	07-10-94 31-01-95 14-01-97